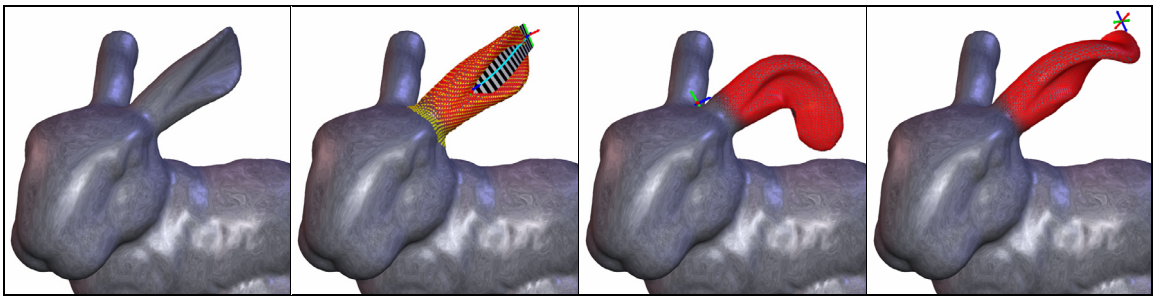

Figure 1: *The initial shape of a bunny's ear (left) is stabbed by a user-controlled ribbon (left center). The user can change the shape of the ear interactively by moving the ribbon, bending it (right center), or twisting it around its centerline wire (right).*

Bender: Deforming 3D Shapes by Bending and Twisting a Virtual Ribbon with both Hands

Ignacio Llamas, Alexander Powell, Jarek Rossignac and Chris Shaw

College of Computing, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Bender is an interactive tool for bending and warping triangulated surfaces. The designer uses a virtual ribbon to grab a portion of the shape and to deform it through direct manipulation. The ribbon is defined by its centerline—a wire made of two smoothly joined circular arcs—and by its twist—the continuous field of normal directions along the wire. The wire and the twist are controlled by a Polhemus tracker in each hand. The deformation model is based on a new formulation of a 3D space warp that uses screw-motions to map coordinate systems aligned with the initial ribbon to corresponding coordinate systems aligned with the final ribbon. Circular biarcs are easy to control and permit the correct handling of situations where a vertex is influenced by different sections of the wire. Screw-motions define smoother and more intuitive warps than other formulations. The combination significantly extends the editing capabilities of previously proposed shape deformation tools and produces smooth and predictable results for configurations where the radius of the tubular region of influence around the ribbon does not exceed the radii of the arcs.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – Curve, surface, solid and object representations; I.3.6 [Computer Graphics]: Methodology and Techniques – Interaction Techniques.

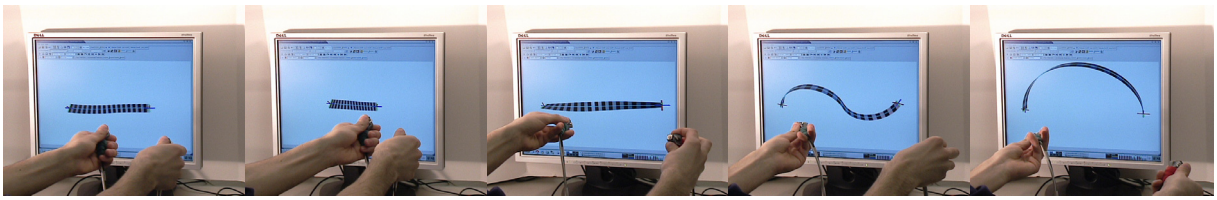

Figure 2: *The user manipulates the two trackers to control the shape of the ribbon.*

1. Introduction

The use of two-handed 3D interaction for sketching and editing 3D shapes has the potential to enhance productivity and artistic freedom for designers. We present an interactive surface deformation tool, Bender, which is not meant to replace existing 3D sculpting tools but complements them by providing unprecedented ease for bending and twisting 3D shapes through direct manipulation.

Like several previously proposed approaches, Bender lets the user control a local space warp which is applied to the vertices of a triangle mesh representation of the surface being edited (Figure 1). In contrast to shape deformations based on space warps that satisfy point displacement constraints [Borrel and Rappoport 1994] and even position and orientation constraints [Gain 2000], [Llamas et al. 2003], Bender makes it easy to bend and twist long protrusions which dominate most animal, organic, and manufactured shapes.

A warp is specified by grabbing a subset of space around a user-controlled virtual ribbon and then changing the shape of the ribbon interactively using a Polhemus [Polhemus 2002] tracker in each hand (see Figure 2). The ribbon is constructed around a central wire. The six degrees of

freedom of each tracker control the position of one end-point of the wire, the direction of the tangent to the wire at this end-point, and the twist of a ribbon around this tangent direction. The concept of the ribbon is used here to capture and communicate to the user how this twist is distributed along the wire (Figure 2). A ribbon that interpolates these two sets of end-conditions is constructed and its image updated as the user keeps moving the trackers. When the user presses a button attached to one of the trackers, the current shape of the ribbon is saved as the initial ribbon. The surface in the vicinity of that initial ribbon will be affected by subsequent motions of the hands. The effect of the deformation of a point will lessen following a user-controlled function of the distance between that point and the initial ribbon. The extent of the Region of Influence (RoI) and the decay function may be quickly adjusted by the user to support large global deformations or the creation of small details. A decay function with a plateau may be used to ensure the preservation of fine details. As the user continues to move the two trackers, the ribbon is moved, rotated, bent and twisted. A space warp that interpolates position, shape, and twist of the initial and the current ribbon is computed and applied in real-time to the surface being edited. This graphic feedback supports the direct manipulation of 3D shapes. When the desired shape is obtained, the user releases the button, hence freezing the warp and saving the new shape for further deformations, if desired.

Each space warp is entirely defined by two pairs of coordinate systems. The initial pair defines an initial ribbon used to grab a portion of the shape when the user presses a button. The final pair is captured when the user releases the button and defines the final ribbon. As in previously proposed approaches [Lazarus et al. 1994], [Singh and Fiume 1998] the vertices of the mesh that lie sufficiently close to the initial ribbon are affected

by the warp.

In addition to providing an effective direct manipulation paradigm, we propose a new representation of the ribbon and a new mathematical model of the warp which offers specific advantages over previous approaches. In particular, the use of a circular biarc for the central wire of the ribbon leads to an intuitive direct manipulation and a very fast computation of all the wire points where the distance to a particular vertex goes through a local minimum. In fact, we prove that only two local minima exist. The use of a warp formulation based on a screw-motion leads to natural shape warps and permits graceful blending of the influences that two distinct regions of the wire may have on a vertex, hence eliminating the tearing problem which occurs when the two vertices of an edge are pulled in different directions by two distinct portions of the wire.

We demonstrate the ease-of-use and power of this formulation in an interactive system called Bender. Although we have not used any spatial indexing to optimize performance, Bender provides 3D graphics feedback at more than 10 frames a second when manipulating surfaces with about 70K triangles. We use adaptive mesh subdivision to refine the surface in areas where the initial tessellation may become visible.

The rest of the paper is organized as follows. In Section 2, we review relevant prior art. In Section 3, we present implementation details and design choices. Finally, we show results and conclude.

2. Related work

A variety of approaches have been followed for creating and changing the shape of a surface more than one vertex at a time. The challenge is to find a pleasing, predictable and controllable method that can be computed in real-time. Some approaches construct surfaces that interpolate 2D profiles [Igarashi et al. 1999] or 3D curves [Wesche and Seidel 2001], [Grossman et al. 2002]. Others provide means for the direct drawing of surfaces [Schkolne et al. 2001] or for space painting and carving [Galyean and Hughes 1991]. An alternative to these shape creation techniques is the warping or deformation of existing shapes. Various methods and interaction paradigms have been developed for this purpose. Sederberg and Parry [1986] introduced the free-form deformation (FFD), based on lattices of control points and trivariate Bernstein polynomials. Hsu et al. [1992] developed a version of FFD that allows direct manipulation, while Coquillart [1990] and MacCracken and Joy [1996] extended FFD to support more general lattices. The technique described in this paper belongs to this group. It is based on a grab-and-drag shape deforming operator, allowing the direct manipulation of shape. It does not limit the user's interaction to control points and does not restrict the operations to be axial deformations.

Based on a designer's natural knowledge of the physical world, we strive to approximate material properties such as elasticity or plasticity. See [Metaxas 1996], [Gibson and Mirtich 1997] and [Gain 2000] for reviews. However, simulated physical realism is generally too expensive for real-time feedback. We have thus opted for a compromise, which offers a simple and intuitive map between hand-gestures and space warps that is independent of the manipulated surface. The cost of computing the warp parameters is negligible and its effect appears physically plausible and quite predictable. Space warping and morphing techniques are thoroughly reviewed by [Gomes et al. 1999].

In the spirit of Forsey and Bartels' Dragon editor [1988], Zorin et al. [1997] presented a system for multiresolution mesh editing in which vertices at different levels of subdivision can preserve details by using adjustment vectors defined in local frames. Other approaches are based on the idea of space warping, which is independent of the representation of the underlying geometry. Barr [1984] introduced the general space deformations twist, bend and taper. Chang and Rockwood [1994] used a generalized de Casteljau approach to extend Barr's technique.

Allan et al. [1989] and Bill [1994] developed systems that displaced a selected vertex and its neighbors by a set of decay functions. Modern software packages, such as Discreet 3D Studio Max 4 and 5 [Discreet 2002], also allow weighted manipulation of vertices with an adjustable decay function. Twister [Llamas et al. 2003] uses a pair of 3D trackers to grab two points on or near a surface and to warp space with a weighting function that decays with increasing range from the trackers. The work described in the present paper can be viewed as an extension of this approach. It is particularly useful for bending long shapes and for operating on elongated regions of influence.

Borrel and Bechmann [1991] and Borrel and Rappoport [1994] developed real-time techniques for computing space warps that simultaneously interpolate several point-displacement constraints. Previous work by Fowler [1992], Gain [2000] and Llamas et al. [2003] support not only point

displacement constraints, but also orientation constraints on points. Milliron et al. [2002] recently introduced a general framework for geometric warps.

The Axial Deformations of Lazarus et al. [1994] used piecewise linear curves of any shape as the axis for a generalized cylinder with variable radii and local frames at key points. Wires, by Singh and Fiume [1998], takes curve based deformation techniques further, but at a higher computational cost. Balakrishnan et al's ShapeTape [1999] uses B-spline curves to create surfaces and Wires to deform shapes using a 3D tracked and instrumented flexible rubber tape.

Turk and O'Brien [2002] approach shape modeling by constructing an implicit surface from scattered data points and normals. Several authors have developed techniques for computing piecewise polynomial surfaces that interpolate points and curves in position and possibly orientation [Hoppe et al. 1992], [Carr et al. 1997], [Bajaj et al. 1995].

Since designers are naturally capable of operating in 3D space, and since 3D surfaces are to be manipulated, we chose to explore a shape operator that provides a natural control of position and orientation of selected regions of space. We justified this decision on the basis of a well-understood interaction style [Shaw and Green 1997], [Hinckley et al 1994] and readily-available hardware [Polhemus 2002]. Using two hands allows the user to adopt both asymmetric [Guiard 1987] and symmetric operations with both hands on the surface being edited. Asymmetric operations allow the dominant hand to adjust fine detail while the non-dominant hand sets up context (position and orientation of the workpiece). Symmetric operations allow each hand to create shapes with its 6 DoF cursor. Offering natural control over six degrees of freedom per hand simplifies the design of complex warps, which will otherwise require a laborious series of 2 DoF or 1 DoF operations if only a mouse is available. Other user interface issues with high degree-of-freedom input devices are explored in Grossman et al. [2003].

3. Implementation details

In this Section, we describe how a central line of the ribbon is computed to interpolate the position and tangent directions at the end-points. Instead of using a cubic parametric curve to solve this Hermite interpolation problem, we use a biarc curve [Rossignac and Requicha 1987] made of two smoothly joined circular arcs. Then, we explain how the additional twist imposed by the two trackers is interpolated along the central line and how it defines a ribbon, and hence defines a coordinate system at each point along the ribbon. We then show how the projection of an arbitrary vertex onto the biarc may be computed efficiently and discuss the key fact that the distance between a point and the biarc may have at most two local minima. Each projection defines a coordinate system on the initial ribbon and the corresponding coordinate system on the final ribbon. To avoid the tearing of space, we compute both projections and use a weighted average of the warps they each define. Then, we describe how a rigid-body screw motion may be computed to interpolate between a starting and an ending coordinate system and also how we combine the effects of two such screw motions. We discuss different weighting functions which, based on the distance between the vertex and its projection on the biarc, determine how much of the rigid body motion will be applied to the vertex. In particular, we discuss the merit of a function with a plateau for preserving the shape of local details. We then discuss techniques that we have developed for ensuring continuity, smoothness, and compactness of the warp along the ribbon. Finally, we provide a brief discussion of the simple strategy we use to perform an adaptive subdivision of the surface where it is required by the extent of the warp.

3.1. Notation

To clarify our notation, consider Figure 3. The wire is a space curve, completely defined by the positions and tangent directions of its two ends (P_0,T_0) and (P_1,T_1). The wire is parameterized by a scalar s in [0,1]. At every point P_S of the wire, T_S denotes the unit tangent to the wire. We consider the wire to be the centerline of a thin piece of surface that we call a ribbon. At every point P_S of the wire, we have one degree of freedom (which we call twist) for rotating the normal N_S to the ribbon's surface around the wire's tangent T_S with respect to the local Frenet coordinate system. This twist is designed to provide a smooth field of normal directions as a linear interpolation between the user-controlled twists at the two ends of the wire. The point P_S and the two unit vectors, T_S, and N_S, suffice to define a local coordinate system C_S at P_S that follows the ribbon in position and orientation as s varies from 0 to 1.

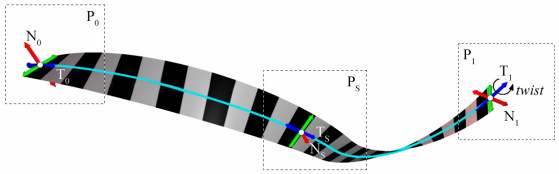

Figure 3: *By specifying the six degrees of freedom at each coordinate system at the end of the wire, (P0, T0, N0) and (P1, T1, N1), the user controls the shape of the wire (cyan) and the orientation (twist) of the ribbon around it. A parameter s defines a point PS on the wire and two orthogonal vectors, TS and NS.*

Hence, for each vertex P of a triangle mesh we compute how the warp affects P. We first compute the projections Q_i of P onto the initial wire. These projections are points on the wire at which the distance to P goes through a local minimum. For all Q_i that are closer to P than a user-prescribed threshold, we compute a displacement vector W_i. The displacement vector W_i is the result of moving P by a fraction f_i of a screw motion M_i. f_i is computed as a function of the distance $\|PQ_i\|$.

The screw motion M_i is computed as follows. From the position of Q_i along the wire, we compute the corresponding parameter s. Then, we compute the corresponding coordinate systems C_s and C'_s on the initial and final wire. M_i is defined as the unique minimal screw motion interpolating between them. We apply a fraction f_i of M_i to P and compute the displacement vector W_i.

3.2. Wire construction

The wire is defined by the positions and tangent directions of its two ends (P_0,T_0) and (P_1,T_1). We wish to create a smooth 3D curve that interpolates the end-conditions and is formed by two circular arcs that are smoothly joined at some point J. The wire is completely defined by computing two scalars, a and b, which define

the points $I_0=P_0+aT_0$ and $I_1=P_1-bT_1$ such that $||I_0I_1||=a+b$, as shown in Figure 4.

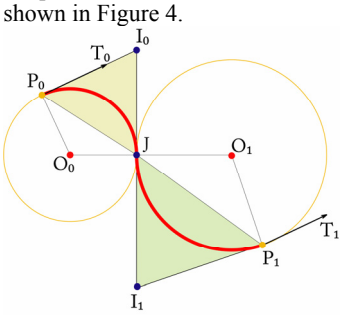

Figure 4

Consider the two points $I_0=P_0+aT_0$ and $I_1=P_1-bT_1$ with a and b chosen so that $||I_0I_1||=a+b$. Consider the point $J=(bI_0+aI_1)/(a+b)$. The triangle (P_0,I_0,J) is isosceles and inscribes a first circular arc that starts at P_0 and is tangent to T_0 and that ends in J and is tangent to I_0I_1. Similarly, the triangle (J,I_1,P_1) is isosceles and inscribes a second circular arc that starts at J where it is tangent to I_0I_1 and ends at P_1 with a tangent to T_1. Both arcs meet at J with a common tangent. Although for clarity Figure 4 was drawn in the plane, the construction holds in three dimensions, when the two triangles are not coplanar. To obtain an example of a 3D situation, simply fold the paper along the I_0I_1.

Following [Rossignac and Requicha 1987], we chose $a=b$. This choice leads to an efficient calculation and yields excellent shapes for the wire. In fact, in most situations the biarc is very close to a cubic parametric curve with the same end-conditions.

To compute the parameter a, we must solve $||(P_0+aT_0)-(P_1-aT_1)||=2a$, which yields a second degree equation in a: $S^2-2a(S\bullet T)+a^2(T^2-4)=0$, where $S=P_1-P_0$ and $T=T_0+T_1$.

In the general case, when $T^2\neq4$, we use $a=-(\text{sqrt}((S\bullet T)^2+(S^2)(4-T^2))-(S\bullet T))/(4-T^2)$, which produces arcs of less than 180 degrees. In the special case where $T^2=4$ and $T_1=T_2$, we use two semi-circles, as discussed in [Rossignac and Requicha 1987].

3.3. Distributing the twist of the ribbon along the biarc

Each arc lies in a plane. The left-hand tracker defines the normal N_0 to the ribbon at P_0. We record the angle a_0 between N_0 and the normal N'_0 to the plane of the first arc. Similarly, we record the angle a_1 between the normal N'_1 to the plane of the second arc and N_1. Let e denote the angle between N'_0 and N'_1. If we rotate P_0 to follow the first arc, and wished to keep the associated normals N_0 and N'_0 in constant orientation with respect to the local Frenet trihedron of the first arc, we would arrive at J with both normals parallel to the plane through J and orthogonal to I_0I_1. Similarly for N_1 and N'_1. In this final configuration, the four normals, N_0, N'_0, N_1, and N'_1 are coplanar and their relative orientations are given by the three angles a_0, a_1, and e. In particular, the angle between N_0 and N_1 is then a_0+e+a_1. Because we wish to obtain a smooth field of normals starting at N_0 and finishing at N_1, we distribute the difference linearly, and twist the local coordinate system C_s along the tangent T_s by an angle equal to $s(a_0+a_1+e)$, where the parameter s is equal to 0 at C_0 and equal to 1 at C_1. In practice, for points on the first arc, we twist C_0 by $s(a_0+a_1+e)$ and then rotate it around the axis of the first arc. For points on the second arc, we rotate C_1 by $(1-s)(a_0+a_1+e)$ and then rotate it backward around the axis of the second arc. Figure 5 shows the ribbon for the same wire with and without twisting.

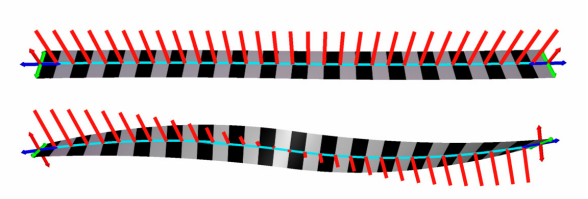

Figure 5: *The ribbon (top) is twisted by rotating both trackers around the tangents to the wire at the two ends (bottom).*

The choice of the s parameterization affects not only the twist of the ribbon around its wire, but also the correspondence between two wires, and hence the warp. We have explored two parameterizations. The first one is an arc-length parameterization for the whole biarc. The second one uses an arc-length parameterization for each arc, forcing $s=0.5$ at the junction.

A parameterization that maps the first arc of the initial wire onto the first arc of the final wire is slightly simpler to compute. However it tends to produce a non-uniform stretching of space when the arc-length ratios between the first and the second arc differ significantly in each wire. We have therefore opted to use the global arc-length parameterization.

3.4. Projection of points onto a biarc

Consider the biarc of the initial ribbon and a point P. As argued above, we want to compute all points Q_i on the biarc where the distance between P and the biarc goes through a local minimum. We will call them the projections of P. We explain in this section how to compute these projections quickly and prove that when P is closer to the wire than the minimum of the bi-arc radii, at most two such projections exist.

Consider a circle with center O, radius r, and normal N. Let Q be the point on the circle that is closest to P. We compute Q by first computing the normal projection $R=P-(PO\bullet N)N$ of P onto the plane of the arc. Then, Q is obtained by displacing O by r towards R. Hence $Q=O+rOR/||OR||$. If Q lies inside the arc, it is a projection of P. Note that if such a normal projection exists on the arc, then all other points of the arc lie further away from P, including the endpoints of the arc. When Q is not on the arc, we consider the free end of the arc, the end of the biarc, as a candidate projection of P. If, at that free end, the biarc moves away from P, then it is a projection of P, i.e., a local minimum of the distance. (Notice that if the other end of the arc were closer to P, it would not be the local minimum for the biarc, since by sliding by an infinitely small amount onto the other arc, it would approach P.)

An example where P has a normal projection inside one arc and on the endpoint of the other arc is shown in Figure 6.

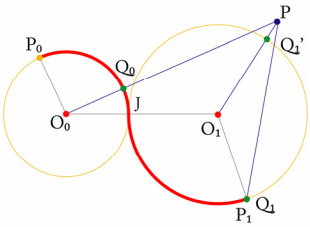

Figure 6: *A closest projection Q0 of P lies inside the first arc. A second closest projection Q1 lies at the tip of the second arc.*

When two projections are returned and both are further away from P than the radius of the region of influence of the warp, P is not affected by the warp. When a single projection is close enough to P, we compute its parameter *s* on the initial ribbon's biarc and use *s* and ||PQ|| to compute a warp. In the cases when the projections Q_0 and Q_1 reported for both arcs are within tolerance from P, we compute two warps and blend them. The merit of this solution is discussed below.

3.5. Deforming a point using screw motion

Given the projection Q of P onto the first arc of the initial wire, we compute its parameter *s* using the ratio of angles (O_0P_0,O_0Q) and (O_0P_0,O_0J) and the ratio of the arc-length of both arcs. A similar approach is used when Q lies on the second arc. We then compute the two coordinates systems, C_s on the initial wire and C_s' on the final wire. They are used as input to compute a fixed point A, an axis direction K, a total rotation angle β, and a total displacement *d*. These four parameters define a screw motion that transforms C_s into C_s' by performing a translation by *d*K and a rotation around the axis (A,K) of an angle β. The computation of these parameters is inexpensive and easily accessible (for example see [Rossignac and Kim 2001], [Llamas et al. 2003]). It will not be repeated here.

We also compute the weight f=F(||PQ||/R), where R is the threshold delimiting the radius of influence around the wire. We discuss the nature of the decay function F below. Then we compute the warped version P' of P by applying to P a translation by fdK and a rotation by angle $f\beta$ around the axis of the screw that has direction K and passes through A. In cases where two projections Q_i fall inside the region of influence, we adjust if necessary the corresponding weights f_0 and f_1, as proposed in [Llamas et al. 2003], compute the images P_0' and P_1' of P through both adjusted warps, and add the displacements they each suggest by moving P to (P_0'+P_1'–P).

3.6. Preventing the tearing of space

When there are two projections Q_0 and Q_1 on the arc where the distance to P is locally minimal and when both fall within the Region of Influence of the initial wire, we must take them both into account. Otherwise, a tearing of space may occur. To explain the tearing, suppose that points P and P' are the endpoints of an edge of the mesh. Suppose that the *s* parameter of the closest projection Q of P is very different from the *s*' parameter of the closest projection Q' of P'. If we were to use the screw associated with a single projection, we would use similar fractions (decay weights) *f*, for both P and P', but their screws could be very different and may pull them away if, for example, the final wire increases the distance between Q and Q'. The edge PP', and hence the incident triangles will be stretched. The corresponding tearing of space is shown in Figure 7.

An important contribution reported here eliminates this tearing. To achieve this, we report two projections, Q_0 and Q_1 for P and two projections Q_0' and Q_1' for P'. We compute the two corresponding screws, M_0 and M_1, for P and blend them as explained below. Similarly, we compute the two screws, M_0' and M_1', for P' and blend them. If P is close to P', then in general Q_0 is close to Q_0' and therefore M_0 is not very different from M_0'. Similarly, Q_1 is close to Q_1' and therefore M_1 is not very different from M_1'. The corresponding weights are also not very different. Consequently, the blended warps will be similar. Hence P and P' will stay close to each other, as illustrated in Figure 7.

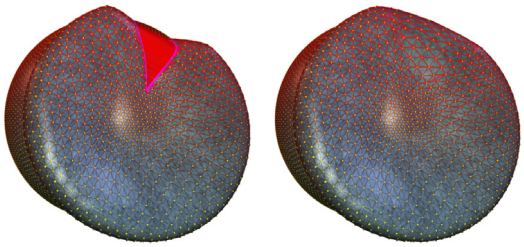

Figure 7: Grabbing a sphere with an initial wire that forms a nearly closed circle and pulling it out and opening the circle produces a tear on the surface (flat region near the top of the left figure). The corrected warp, based on the use of two projections, is shown on the right.

3.7. Choosing decay functions

Depending on the type of deformation we want to achieve, different decay functions F may be preferred. Following [Jin 2000], [Lazarus et al. 1994], we let the user switch between a bell-shaped curve and a plateau function (Figure 8), which permits to preserve the shape inside a tube around the wire when the relation between the corresponding portion of the initial and final ribbons is a rigid body transformation. Such relations are maintained when performing warps that achieve rigid bending operation of limbs or tubes.

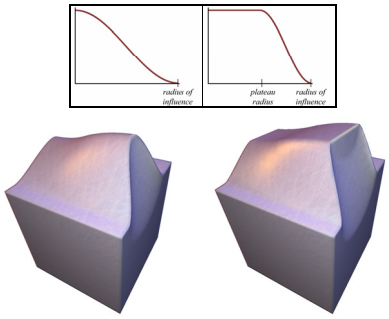

Figure 8: *We show the results produced by using the two decay functions. The top shows a profile view of the decay functions. The initial and final wires were identical in both cases.*

3.8. Maintaining continuity

In this subsection, we discuss a modification to the screw computation, which was necessary to ensure the continuity of the warp through space.

The screw motion interpolation used in [Llamas et al. 2003] always generates screws of minimal angle, which is always less than 180 degrees. Consider two points P_S and P_S' traveling simultaneously on the initial and final wire. Assume that they move towards each other, then go through a singular situation where their velocities are parallel, and finally diverge. As we pass through the singular situation, the orientation of the screw axis K is reversed. The displacement values and the direction of rotation are also reversed. This flip produces a discontinuity in the pencil of helix trajectories taken by points of the initial wire as they are warped (Figure 9). We detect these situations using the sign of the dot product of consecutive K vectors. To prevent the discontinuity, we simply revert the flip. This correction results in rotation angles β that may temporarily exceed 180 degrees. We compute the angle as before, and simply replace it by $(\beta–2\pi)$. The K axis is reversed and the distance d negated. We do this change at each singular point.

When no correction is needed, we use the natural direction of K given by the original construction in [Llamas et al. 2003]. When one or more corrections are needed, the user may press a button to toggle between the two possibilities, the one defined at s=0 by the original construction of K and the one where all the K directions are reversed.

Note however, that neither the flip of K nor the blending of screws associated with two projections will solve the problem of space inversion that is inherent to all wire-based warps and may occur when the radius of the region of influence is larger that the minimum radius of curvature of the wire. Modeling the wire as a biarc makes it trivial to detect these situations because the radius of curvature is known for each arc. Thus, we have considered reducing the radius of the region of influence automatically to avoid such inversions. However, because undesirable space inversions are easy to detect visually and avoid with direct manipulation, we have opted not to perform the automatic adjustment to avoid surprising the user with the occasional incorrect choice.

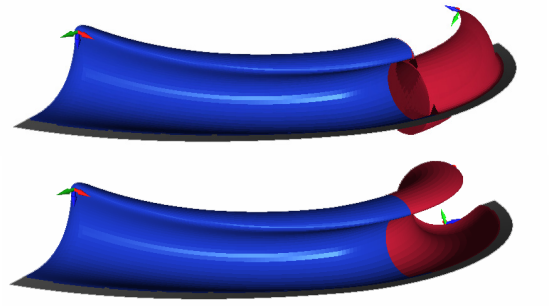

Figure 9: The surface (top) that interpolates the two wires is swept by the helix trajectory followed by a wire point P_S when it is moved by the corresponding screw C_S to its destination. The undesired bulges are removed (bottom) by preventing sudden flips of the screw axis direction K and by permitting for the screw motion to have an angle of more than 180 degrees.

3.9. Adaptive subdivision

When the mesh is stretched by a warp, the density of its tessellation may no longer be sufficient to produce a smooth warped surface (Figure 10). We use a simple and very efficient technique for adaptively subdividing the surface wherever appropriate. After each warp, when the user freezes the shape, the system starts an adaptive subdivision process and replaces the warped surface with a smoother one. Note that our subdivision simply splits some triangles into 2, 3, or 4 smaller triangles without changing the initial shape. Contrary to subdivision procedures that smoothen the shape, in our implementation, the new vertices are positioned exactly in the middle of the old edges and the old vertices are not adjusted. Tucking in of the old vertices as a Loop subdivision would do or bulging out the edges as a Butterfly subdivision would do is unnecessary, if the initial shape was sufficiently smooth. Hence, we do not have to respect restrictions on the subdivision levels between neighboring triangles.

Let the term initial mesh denote the mesh before the current warp, which deforms it into a final mesh. Note that the initial mesh may have been produced by a series of previous warps and subdivisions. Each edge of the initial mesh is tested and marked if subdivision is required. Then, each marked edge is split at its mid-point and each triangle with m marked edges is subdivided into m+1 triangles, using a standard split. This simple approach guarantees preservation of connectivity and does not introduce T-junctions. To test whether an edge should be marked, we compute the distance between the mid-point of its warped vertices and the warped midpoint of its vertices. If that distance exceeds a threshold, we mark the edge. The process is repeated until no more edges need to split or until the user starts a new warp.

This simple approach works well in practice and is very fast. However, it does not guarantee detection of all cases where subdivision is needed. For example, a local stretch occurring inside a triangle that does not affect the edge-midpoints could remain undetected.

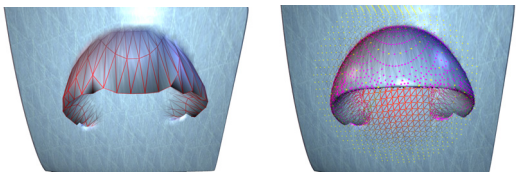

Figure 10: *A surface has been warped using its original triangulation (left). A smoother surface is produced through an adaptive subdivision (center, right).*

4. Concluding remarks

By combining a biarc wire with the concept of a twisted ribbon around it and with a screw-based motion that interpolates corresponding portions of the initial and final ribbons, we have created a new formulation of a space warp that is completely defined by four coordinate systems. We have developed a prototype 3D user interface for the direct manipulation of these coordinate systems through the use of

two Polhemus trackers. We show that the approach makes it easy to design or bend, twist, or warp a variety of shapes.

Before opting for this approach, we have explored other formulations for the wire and for the warp. For instance, using a cubic curve or a helix as a wire results in a significantly more expensive calculation of the vertex projections on the wire and could potentially generate a larger number of projections. Hence, we have chosen to use a circular biarc for three reasons. First, it provides the user with a very intuitive control of the shape of the wire. Second, it significantly reduces the cost of computing the projection Q of a point P and the associated coordinate system, when compared to a helix or to a cubic polynomial curve [Schneider 1990]. Third, our choice ensures that there are at most two locally closest projections Q_0 and Q_1 of any point P onto a biarc. Based on this observation, we are able to develop a simple technique for avoiding the tearing of space that happens when two neighboring surface points P and P' have each locally closest projections that are distant along the wire.

We have also explored using motions that are not screw motions for the interpolation between a coordinate system on the initial wire and its counterpart on the final wire. In particular, we have explored the use of a biarc-driven trajectory. We have concluded that the combination presented in this paper is by far the best, producing natural warps, avoiding undesired bulges, and yielding a very fast implementation.

The design choices we made lead to an intuitive and predictable deformation, even when the changes in the shape and twist of the initial and final ribbons are significant. Furthermore, it permits a real-time direct manipulation, even for shapes of significant complexity. For example, our current, unoptimized implementation produces 10 frames per second with models of about 70,000 triangles.

Figure 11 illustrates the variety of shape deformations that may be trivially achieved by a single Bender warp.

Acknowledgement

This work was supported by the NSF under the ITR Digital Clay grant 0121663.

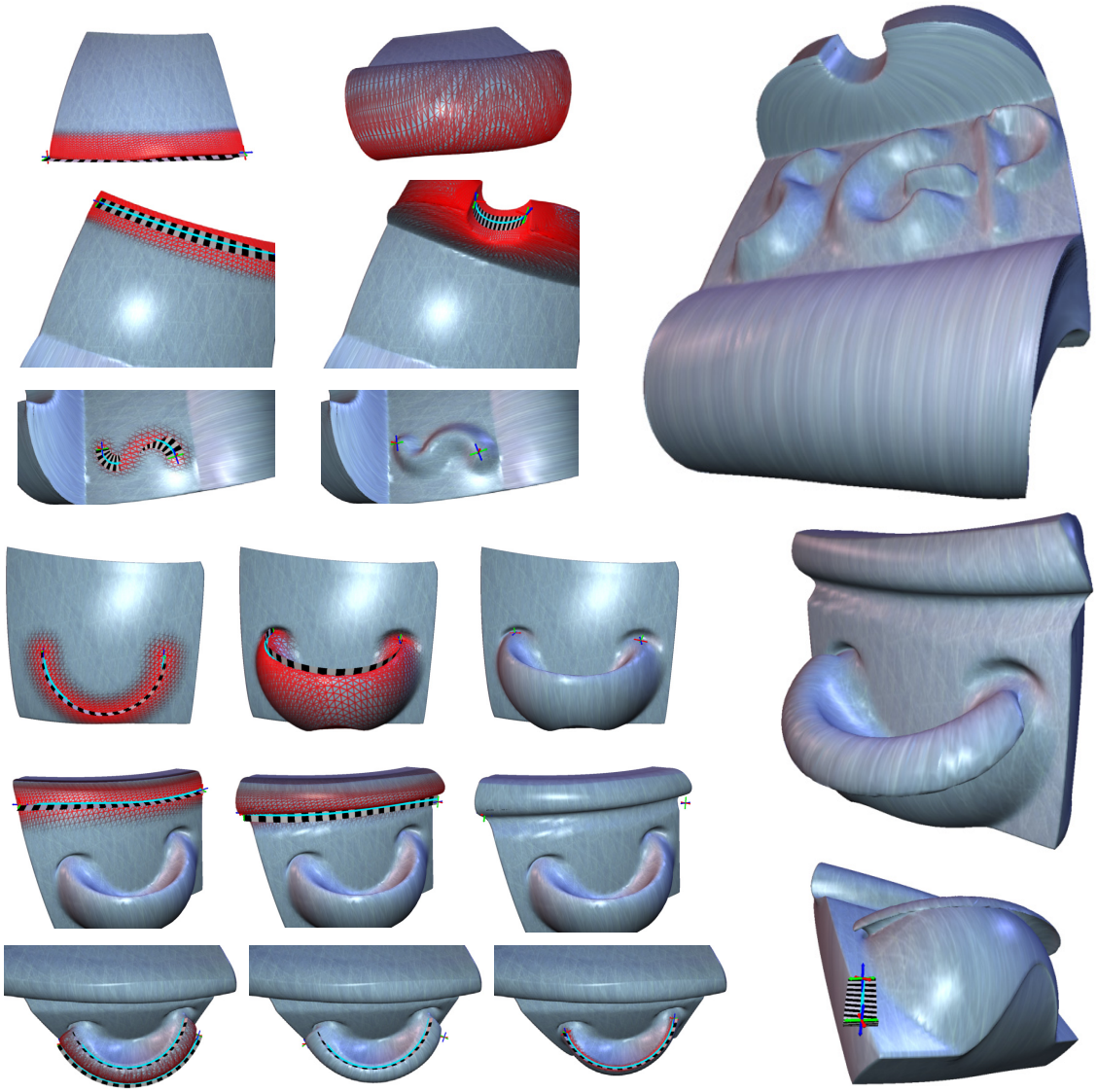

Figure 11: *Sequences of Bender steps used to create two models.*

References

ALLAN, J. B., WYVILL, B., AND WITTEN, I. 1989. A Methodology for Direct Manipulation of Polygon Meshes. *New Advances in Computer Graphics (Proceedings of CG International '89)* (June), 451-469.

BAJAJ, C., CHEN, J., AND XU, G. 1995. Modeling with Cubic A-patches. *ACM Transactions on Computer Graphics 14*, 2, 103-133.

BALAKRISHNAN, R., FITZMAURICE, G.W., KURTENBACH, G., SINGH, K. 1999. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In *Proceeding of the 1999 Symposium on Interactive 3D Graphics*. ACM Press. ACM. 111-118.

BARR, A. H. 1984. Global and Local Deformations of Solid Primitives. In *Computer Graphics (Proceedings of ACM SIGGRAPH 84)*, ACM SIGGRAPH, 21-30.

BILL, J. R., AND LODHA, S. 1994. Computer Sculpting of Polygonal Models using Virtual Tools. Tech. Rep. UCSC-CRL-94-27, Baskin Center for Computer Engineering and Information Sciences, University of California, Santa Cruz, U.S.A.,July.

BORREL, P., AND BECHMANN, D. 1991. Deformation of n-Dimensional Objects. In *SMA '91: Proceedings of the First Symposium on Solid Modeling Foundations and CAD/CAM Applications*, ACM Press, ACM, 351-370.

BORREL, P., AND RAPPOPORT, A. 1994. Simple Constrained Deformations for Geometric Modeling and Interactive Design. *ACM Transactions on Graphics 13*, 2 (April), 137-155.

CARR, J. C., FRIGHT, W. R., AND BEATSON, R. K. 1997. Surface Interpolation with Radial Basis Functions for Medical Imaging. *IEEE Transactions on Medical Imaging 16*, 1 (February), 96-107.

CHANG, Y.-K., AND ROCKWOOD, A. P. 1994. A Generalized de Casteljau Approach to 3D Free-form Deformation. In *Proceedings of ACM SIGGRAPH 1994*, ACM Press, ACM SIGGRAPH, 257-260.

COQUILLART, S. 1990. Extended Free-Form Deformation: A Sculpting Tool for 3D Geometric Modeling. In *Computer Graphics (Proceedings of ACM SIGGRAPH 90)*, ACM SIGGRAPH, 187-196.

DACHILLE, F.I., QIN, H., KAUFMAN, A., AND EL-SANA, J. 1999. Haptic sculpting of dynamic surfaces. In *Proceedings of the 1999 Symposium on Interactive 3D graphics*, ACM Press, ACM, 103-110.

DISCREET, 2002. Discreet 3ds max, http://www.discreet.com/products/3dsmax/.

Forsey, D.R., AND Bartels, R. H. 1988. Hierarchical B-spline refinement. *SIGGRAPH 88*, 205-212.

FOWLER, B. 1992. Geometric Manipulation of Tensor Product Surfaces. In *Proceedings of the 1992 Symposium on Interactive 3D graphics*, ACM Press, ACM SIGGRAPH, 101-108.

GAIN, J. E. 2000. *Enhancing Spatial Deformation for Virtual Sculpting*. PhD thesis, St. Johns College, University of Cambridge.

GALYEAN, T. A., AND HUGHES, J. F. 1991. Sculpting: an Interactive Volumetric Modeling Technique. In *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, ACM Press, ACM SIGGRAPH, 267-274.

GIBSON, S. F. F., AND MIRTICH, B. 1997. A Survey of Deformable Modeling in Computer Graphics. Tech. rep., Mitsubish Electric Research Laboratoy.

GOMES, J., DARSE, L., COSTA, B., AND VELHO, L. 1999. *Warping and Morphing of Graphical Objects*. Morgan Kaufmann Publishers Inc.

GROSSMAN, T., BALAKRISHNAN, R., KURTENBACH, G., FITZMAURICE, G., KHAN, A., AND BUXTON, B. 2002. Creating Principal 3D Curves with Digital Tape Drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, ACM SIGCHI, 121-128.

GROSSMAN, T., BALAKRISHNAN, R. AND SINGH, K. 2003. An interface for creating and manipulating curves using a high degree-of-freedom curve input device. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, ACM SIGCHI.

GUIARD, Y. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *The Journal of Motor Behavior*, 19, 4, 486-517.

HINCKLEY, K., PAUCH, R., GOBLE, J. C. AND KASSEL, N. F. 1994. Passive Real-World Interface Props for Neurosurgical Visualization. In *Human Factors in Computing Systems CHI'94 Conference Proceedings*. 452-458.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface Reconstruction from Unorganized Points. In *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, ACM Press, 71-78.

HSU, W. M., HUGHES, J. F., AND KAUFMAN, H. 1992. Direct Manipulation of Free-Form Deformations. In *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, vol. 26, ACM SIGGRAPH, 177.184.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a Sketching Interface for 3D Freeform Design. In *Proceedings of ACM SIGGRAPH 99*, ACM Press/Addison-Wesley Publishing Co., ACM SIGGRAPH, 409-416.

Jin, X., Li, Y.., AND Peng, Q. 2000. General constrained deformations based on generalized meatballs. In *Computers & Graphics 2000*, Elsevier, 24, 219-231.

LAZARUS, F., COQUILLART, S., AND JANCENE, P. 1994. Axial Deformations: An Intuitive Deformation. *Computer Aided Design 26*, 8 (August), 607-613.

LLAMAS, I., KIM, B., GARGUS, J., ROSSIGNAC, J. AND SHAW C.D. 2003. Twister: a Space-Warp Operator for the Two-Handed Editing of 3D Shapes. *ACM Transactions on Graphics (TOG) 22*, 3, Proceeding of SIGGRAPH 2003.

MACCRACKEN, R., AND JOY, K. I. 1996. Free-Form Deformation With Lattices of Arbitrary Topology. In *Proceedings of ACM SIGGRAPH 1996*, ACM Press, ACM SIGGRAPH, 181-190.

METAXAS, D. N. 1996. *Physics-Based Deformable Models: Applications to Computer Vision, Graphics, and Medical Imaging*. Kluwer Academic Publishers, January.

MILLIRON, T., JENSEN, R. J., BARZEL, R., AND FINKELSTEIN, A. 2002. A Framework for Geometric Warps and Deformations. *ACM Transactions on Graphics (TOG) 21*, 1, 20-51.

POLHEMUS, 2002. Polhemus Fastrak, http://www.polhemus.com/ftrakds.htm.

ROSSIGNAC, J. R., AND KIM, J. J. 2001. Computing and Visualizing Pose-Interpolating 3D Motions. *Computer Aided Design 33*, 4 (April), 279-291.

Rossignac, J.R., AND Requicha, A. A. G. 1987. Piecewise-circular curves for geometric modeling. *IBM J. Res Develop. Vol 31*, 3 (May) 296-313.

SCHKOLNE, S., PRUETT, M., AND SCHRÖDER, P. 2001. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, ACM SIGCHI, 261-268.

Schneider, P. 1990. Solving the Nearest-Point-on-Curve Problem. *Graphics Gems,* Academic Press, vol.1: 607–612.

SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-Form Deformation of Solid Geometric Models. In *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, ACM SIGGRAPH, 151-160.

SHAW, C., AND GREEN, M. 1997. THRED: A Two-Handed Design System. *Multimedia Systems 5*, 2, 126-139.

SINGH, K., AND FIUME, E. 1998. Wires: A Geometric Deformation Technique. In *Proceedings of ACM SIGGRAPH 1998*, ACM Press, ACM SIGGRAPH, 405-414.

TURK, G., AND O'BRIEN, J. F. 2002. Modelling with Implicit Surfaces that Interpolate. *ACM Transactions on Graphics 21*, 4 (October), 855-873.

WESCHE, G., AND SEIDEL, H.-P. 2001. FreeDrawer: a Free-Form Sketching System on the Responsive Workbench. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ACM Press, ACM, 167-174.

ZORIN, D., SCHRÖDER, P., AND SWELDENS, W. 1997. Interactive Multiresolution Mesh Editing. In *Proceedings of ACM SIGGRAPH 1997*, ACM Press, ACM SIGGRAPH, 256-268.